Amendments to the claims,
Listing of all claims pursuant to 37 CFR 1.121(c)

This listing of claims will replace all prior versions, and listings, of claims in the application:

What is claimed is:

1. (Currently amended) In a computer system operating under control of an operating system, a method for ~~controlling~~ detecting and preventing indirect access to a computer network by potentially malicious applications already installed and executing on the computer system, the method comprising:

defining rules indicating which system services of the operating system are monitored for detecting and preventing indirect access to the computer network by potentially malicious applications that are already installed and executing on the computer system, but which are capable of obtaining indirect access to the computer network through system services;

trapping an attempt by a particular application already installed and executing on the computer system to gain indirect access to the computer network ~~to invoke~~ through invocation of a particular system service being monitored;

~~determining~~ detecting if the attempt to invoke the particular system service constitutes an unauthorized attempt by ~~an unauthorized~~ a potentially malicious application already installed and executing on the computer system to obtain indirect access to the computer network by invoking the particular system service which in turn accesses the computer network on behalf of the ~~unauthorized~~ potentially malicious application; and

if the attempt to invoke the particular system service constitutes an unauthorized attempt by ~~an unauthorized~~ a potentially malicious application to access the computer network indirectly, preventing the potentially malicious application from obtaining indirect access to the computer network by blocking the attempt.

2. (Original) The method of claim 1, wherein said trapping step includes intercepting operating system calls for invoking the particular system service.

3. (Original) The method of claim 1, wherein said trapping step includes intercepting local procedure calls for invoking the particular system service.

4. (Original) The method of claim 1, wherein said trapping step includes intercepting an attempt to open a communication channel to the particular system service.

5. (Original) The method of claim 1, wherein said trapping step includes rerouting an attempt to invoke the particular system service from a system dispatch table to an interprocess communication controller for determining whether to block the attempt based on the rules.

6. (Original) The method of claim 5, wherein said step of rerouting attempts to invoke the particular system service from a dispatch table to the interprocess communication controller includes replacing an original destination address in the system dispatch table with an address of the interprocess communication controller.

7. (Original) The method of claim 6, further comprising the steps of:

retaining the original destination address; and

using the original destination address for invoking the particular system service if the interprocess communication controller determines not to block the attempt.

8. (Previously presented) The method of claim 1, wherein the rules specifying which system services are monitored are established based on user input.

9. (Previously presented) The method of claim 1, wherein the step of blocking the attempt is based upon consulting a rules engine for determining applications authorized to invoke the particular system service.

10. (Currently amended) The method of claim 1, wherein the step of ~~blocking the attempt~~ preventing includes obtaining user input as to whether the unauthorized application should now be authorized to invoke the particular system service.

11. (Previously presented) The method of claim 10, wherein said step of obtaining user input includes the substeps of:

providing information to the user about which particular application is attempting to invoke the particular system service; and

receiving user input as to whether the particular application should be blocked from invoking the particular system service.

12. (Original) A computer-readable medium having computer-executable instructions for performing the method of claim 1.

13. (Previously presented) The method of claim 1, further comprising:

downloading a set of computer-executable instructions for performing the method of claim 1.

14. (Currently amended) In a computer system operating under control of an operating system, a method for ~~regulating~~ detecting and preventing indirect access to the Internet by potentially malicious applications already installed and executing on the computer system, the method comprising:

defining a policy indicating which system services of the operating system are monitored for detecting and preventing indirect access to the Internet by potentially malicious applications that are already installed and executing on the computer system but which are capable of obtaining indirect access to the Internet through system services, said policy specifying processes authorized to access the Internet;

intercepting an attempt by a first process to communicate with a second process in a manner that provides the first process with indirect access to Internet;

identifying the first process that is attempting to communicate with the second process;

identifying the second process;

based on said policy, determining whether the first process may communicate with the second process in a manner that provides the first process with indirect access to

Internet, including determining if the attempt to invoke the particular system service does constitutes an unauthorized attempt by a potentially malicious application to access the Internet indirectly; and

allowing the first process to communicate with the second process if said policy indicates that the first process may communicate with the second process in a manner that provides the first process with indirect access to Internet and does not constitutes an unauthorized attempt by a potentially malicious application to access the Internet indirectly.

15. (Original) The method of claim 14, wherein the first process comprises an instance of an application program.

16. (Original) The method of claim 14, wherein the second process comprises a system service.

17. (Original) The method of claim 14, wherein said intercepting step includes intercepting operating system calls made by the first process to attempt to communicate with the second process.

18. (Original) The method of claim 14, wherein said intercepting step includes detecting local procedure calls.

19. (Original) The method of claim 14, wherein said intercepting step includes detecting an attempt by the first process to open a communication channel to the second process.

20. (Original) The method of claim 14, wherein said intercepting step includes rerouting attempts by the first process to communicate with the second process from a system dispatch table to an interprocess communication controller.

21. (Original) The method of claim 14, wherein said step of identifying the

second process includes evaluating parameters of the attempt made by the first process to communicate with the second process.

22. (Original) The method of claim 14, wherein said policy specifies particular processes to be protected from communications made by other processes.

23. (Original) The method of claim 14, further comprising:

providing for a process to be registered in order to be protected from communications made by other processes; and

determining whether to allow the first process to communicate with the second process based, at least in part, upon determining whether the second process is registered.

24. (Original) The method of claim 23, wherein said determining step is based, at least in part, on the type of communication the first process is attempting with the second process.

25. (Currently amended) In a computer system operating under control of an operating system, a method for detecting and preventing one application already installed and executing on the computer system from gaining indirect Internet access through other applications, the method comprising:

registering a first application to be protected from serving as a proxy by which other applications may gain indirect Internet access for detecting and preventing indirect access to the Internet by potentially malicious applications that are already installed and executing on the computer system but which are capable of obtaining indirect access to the Internet through said first application;

detecting an attempt by a second application already installed and executing on the computer system to access the first application for purposes of using the first application as a proxy for indirect Internet access;

identifying ~~a~~ the second application that is attempting to access the first application for purposes of using the first application as a proxy for indirect Internet access; and

rerouting the attempt to access the first application through an interprocess communication controller that determines whether to allow the attempt, based on rules indicating whether the second application is authorized to access the first application using interprocess communication.

26. (Original) The method of claim 25, wherein said registering step includes supplying rules specifying particular communications from which the first application is to be protected.

27. (Original) The method of claim 26, wherein the interprocess communication controller determines whether to allow the attempt based, at least in part, upon the rules specifying particular communications from which the first application is to be protected.

28. (Original) The method of claim 25, wherein said detecting step includes intercepting operating system calls for accessing the first application.

29. (Original) The method of claim 25, wherein said detecting step includes detecting a graphical device interface (GDI) message sent to the first application.

30. (Original) The method of claim 29, wherein said identifying step includes evaluating parameters of the message sent to the first application.

31. (Original) The method of claim 25, wherein said detecting step includes detecting an attempt to send keystoke data to a window of the first application.

32. (Original) The method of claim 25, wherein said detecting step includes detecting an attempt to send mouse movement data to a window of the first application.

33. (Original) The method of claim 25, wherein said rerouting step includes rerouting the attempt to access the first application from a system dispatch table to the interprocess communication controller.

34. (Original) The method of claim 25, wherein said rules indicating whether the second application may access the first application includes rules indicating particular types of communications which are allowed.

35. (Original) The method of claim 25, further comprising:

if the interprocess communication controller allows the attempt to access the first application, routing the attempt to the first application.

36. (Currently amended) A system for ~~regulating~~ detecting and preventing indirect access to the Internet access by controlling interprocess communication between applications, the system comprising:

a computer having at least one processor, said computer operating under control of an operating system providing interprocess communication;

a policy specifying applications that are permitted to communicate with a first application using interprocess communication, said first application capable of providing indirect Internet access to other applications, so as to detect and prevent indirect access to the Internet by potentially malicious applications that are already installed and executing on the computer system but which are capable of obtaining indirect access to the Internet through said first application;

a module for detecting an attempt by a second application already installed and executing on the computer system to gain indirect Internet access through the first application using interprocess communication; and

an interprocess communication controller for identifying the second application attempting to gain indirect Internet access through the first application using interprocess communiation and determining whether to permit the communication based upon the identification of the second application and the policy specifying applications permitted to communicate with the first application.

37. (Original) The system of claim 36, wherein said policy includes rules indicating particular types of communications which are permitted.

38. (Original) The system of claim 36, further comprising:

a rules engine for specifying applications that are permitted to communicate with the first application using interprocess communication.

39. (Original) The system of claim 36, further comprising:

a registration module for establishing said policy.

40. (Original) The system of claim 39, wherein said registration module provides for identifying applications to be governed by said policy.

41. (Original) The system of claim 36, wherein said module for detecting a second application detects an operating system call to open a communication channel to the first application.

42. (Original) The system of claim 36, wherein said module for detecting a second application detects a graphical device interface (GDI) message sent to the first application.

43. (Original) The system of claim 36, wherein said module for detecting a second application detects a local procedure call attempting to access the first application.

44. (Original) The system of claim 36, wherein said module for detecting a second application redirects attempts to communicate with the first application to the interprocess communication controller.

45. (Original) The system of claim 36, wherein said module for detecting a second application reroutes the attempt to communicate with the first application from a dispatch table to the interprocess communication controller.

46. (Original) The system of claim 36, wherein said interprocess communication

controller determines whether to permit the communication based, at least in part, upon evaluating parameters of the attempt made by the second application to communicate with the first application.

47. (Original) The system of claim 36, wherein said interprocess communication controller determines whether to permit the communication based upon obtaining user input as to whether to permit the second application to communicate with the first application.